



**Vasco Samuel Afonso
Gomes**

**Desenvolvimento de um sistema de visão
computacional para drones comerciais**



**Vasco Samuel Afonso
Gomes**

**Desenvolvimento de um sistema de visão
computacional para drones comerciais**

Dissertação apresentada à Universidade de Aveiro em cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Automação Industrial, realizada sob a orientação científica do Professor Doutor António José Ribeiro Neves, docente do Departamento de Electrónica Telecomunicações e Informática da Universidade de Aveiro

o júri / the jury

presidente / president

Professor Doutor Vítor Manuel Ferreira dos Santos

Professor Associado da Universidade de Aveiro

vogais / examiners committee

Doutora Alina Liliana Trifan

Bolseira de Pós-Doutoramento do Instituto de Engenharia Eletrónica e Telemática de Aveiro (Arguente)

vogais / examiners committee

Professor Doutor António José Ribeiro Neves

Professor Auxiliar da Universidade de Aveiro (orientador)

**agradecimentos /
acknowledgements**

Em Primeiro lugar, agradecer aos meus familiares, pais e irmã, pelo apoio e disponibilidade ao longo do meu período académico pois sem eles os objetivos não teriam sido atingidos.

Seguidamente, um agradecimento especial ao orientador, professor António Neves, pela ajuda, apoio e disponibilidade durante a realização desta dissertação.

Palavras Chave

Drones, Visão por computador, detecção de objetos, seguimento visual, OpenCV

Resumo

Drones e visão computacional são duas realidades muito atuais, através das quais é possível obter todo um conjunto de informação e experiências. Conjugando estas tecnologias permite-nos perceber e interagir com o mundo, possibilitando inclusive atuar em situações de risco como os fogos florestais, na procura de pessoas em locais de difícil acesso, ou na monitorização de processos de produção como a agricultura de precisão.

Esta dissertação tem como objetivo o desenvolvimento e integração de um sistema de visão computacional num drone comercial capaz de identificar e seguir um determinado objeto de interesse colorido. Para atingir este objetivo, o trabalho dividiu-se em duas fases. A primeira consistiu no desenvolvimento de algoritmos para a calibração do sistema de visão com vista à obtenção da cor do objeto de interesse a seguir, para que, à posteriori, seja possível distingui-lo numa imagem com outros elementos. A segunda consistiu no desenvolvimento do sistema de seguimento e, consequentemente, a determinação das coordenadas físicas a enviar ao drone para que este seja capaz de seguir de forma autónoma o objeto de interesse.

Com a realização das várias tarefas, foi possível no final obter um protótipo do sistema de visão. Os resultados experimentais obtidos permitem concluir que é vantajoso usar um método de escolha de cor automático e verificar que com o valor de seguimento correto é possível acompanhar o objeto de interesse de acordo com as condições estabelecidas.

Keywords

Unmanned Aerial Vehicles, Computer vision, Object detection, visual tracking, OpenCV

Abstract

Drones and computer vision are two realities where it is possible to get different information and experiences. These technologies allow us to perceive and interact with the world, acting, for example, in risk situations like forest fires, find missing people in difficult places or monitor production processes like precision agriculture.

This thesis has as main objective to develop and to integrate a computer vision system into a commercial drone with the capability of identify and follow colored objects. to achieve this goal, it was necessary the development of two main tasks. One of them was the development of calibration algorithms in order to obtain the color of the object to follow. The second one was the development of the tracking system, and, consequently, estimate the coordinates to send to the drone in order to follow the object.

In the end, a prototype of a computer vision system was developed. Taking into consideration the experimental results, the main conclusions are the advantage of using an automatic method select the colour of interest and verify that with the correct value of tracking it is possible to follow object of interest accordingly to pre defined conditions.

Conteúdo

Conteúdo	i
Lista de Figuras	iii
Lista de Tabelas	v
1 Introdução	1
1.1 Estrutura da Dissertação	1
2 Estado da Arte	3
2.1 Floresta	3
2.2 Monitorização de pessoas	6
2.3 Sistemas de <i>Tracking</i>	7
3 Sistema de Visão	11
3.1 <i>Hardware</i> e Sistema de testes	11
3.2 Calibração de câmara	13
3.3 Seleção da cor de interesse	16
3.3.1 <i>Floodfill</i>	16
3.3.2 Seleção Manual (<i>Slider</i>)	17
3.3.3 Histograma	17
4 Sistema de <i>tracking</i>	19
4.1 Identificação dos objetos coloridos e possíveis alvos	19
4.2 <i>Threshold</i> de <i>tracking</i>	21
4.3 Processo de <i>Tracking</i>	24
4.4 Controlo	26
5 Conclusões e Trabalho futuro	29
5.1 Trabalho Futuro	30

6	Anexo A	31
6.0.1	Caraterísticas Pi Câmera [11]	31
7	Anexo B	33
7.1	Escolha dos objetos por maior área	33
8	Anexo C	37
8.1	Escolha dos objetos pelo utilizador	37
	Bibliografia	41

Lista de Figuras

1.1	Diagrama funcional do sistema de visão computacional que será desenvolvido no âmbito desta dissertação.	2
2.1	Fluxograma do algoritmo proposto em [5].	4
2.2	<i>Layout</i> experimental para monitorização e deteção de incêndios baseado em UAVs [5].	5
2.3	Evolução do processamento para a primeira parte [5].	5
2.4	Evolução do processamento para a segunda parte [5].	5
2.5	Curva de intensidade de sinal em relação à profundidade [13].	6
2.6	Imagem teste para o processo de segmentação de cor [6].	7
2.7	Binarização pelo modelo HSV [6].	8
2.8	Teste à aplicação desenvolvida [6].	8
2.9	Imagem teste para o processo segmentação de cor [6].	9
2.10	Aplicação do algoritmo SURF [6].	9
3.1	Raspberry Pi.	12
3.2	Picamara.	12
3.3	Ambiente de testes criado para validação de resultados.	13
3.4	Modelo pinhole	14
3.5	Processo de calibração de uma câmara.	15
3.6	Escolha da cor a seguir por processo de <i>floodfill</i>	17
3.7	Escolha da cor a seguir por processo de seleção manual (<i>slider</i>).	18
3.8	Escolha da cor a seguir por análise do Histograma	18
4.1	Espaço de cor HSV	20
4.2	Escolha do objeto com maior área a seguir.	21
4.3	Seleção do objeto alvo pelo utilizador.	21
4.4	Diagrama de calculo do <i>Threshold</i> de <i>tracking</i>	22

4.5	Exemplo do processo de análise de regiões.	24
4.6	Exemplo da detecção de dois objetos com a mesma cor	26
4.7	Exemplo do posicionamento do objeto de interesse a seguir.	27

Lista de Tabelas

4.1	Distância entre as posições do objeto a seguir (posição atual e posição anterior)	26
4.2	Resultado da distância entre o objeto e o centro da imagem.	27

Capítulo 1

Introdução

Os veículos aéreos não tripulados (UAVs) ou Drones são dispositivos capazes de voar sem a necessidade de existir um piloto humano a bordo, sendo possível controlá-lo remotamente ou simplesmente configurá-los para voarem de forma autónoma [3]. Do ponto de vista da investigação, este assunto é tema de debate intensivo nos dias de hoje, como se verifica pelo vasto numero de publicações em revistas e conferências da especialidade, como é o caso do evento *IEEE International Conference on Unmanned Aircraft Systems* (IEEE ICUAS), entre muitas outras.

Os vídeos capturados por estas aeronaves são cada vez mais comuns nos nossos dias, sendo possível obter imagens de locais de difícil acesso para um ser humano. Todavia, se a estas imagens se aplicar algum processamento através de um sistema de visão computacional [4], torna-se viável a sua utilização em aplicações concretas, como se comprova pelos diversos projetos e produtos já existentes no mercado com aplicações em áreas como a monitorização de veículos [8], agricultura de precisão [17], os fogos florestais [14] ou situações de catástrofe [2].

Pretende-se com esta dissertação desenvolver um sistema de processamento de imagem e incorporá-lo num drone comercial. Este sistema será desenvolvido com recurso a um microcomputador, que permitirá ter um sistema com capacidade de processamento em tempo real. Como prova de conceito, o objetivo consiste em detetar e seguir um determinado objeto colorido. Uma representação dos principais módulos do sistema a desenvolver pode ser vista na Figura 1.1.

1.1 Estrutura da Dissertação

Esta dissertação encontra-se dividida em três capítulos, para além desta Introdução e Conclusão no final.

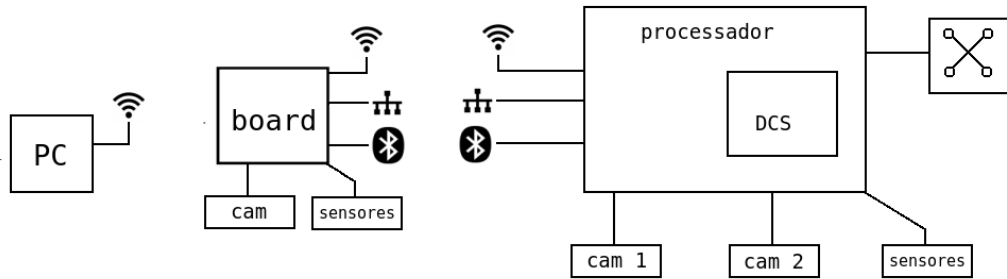


Figura 1.1: Diagrama funcional do sistema de visão computacional que será desenvolvido no âmbito desta dissertação.

No Capítulo 2 é apresentado o estado da arte, ou seja, é descrito o levantamento e estudo realizado acerca do tema abordado neste trabalho. Aqui são apresentadas algumas aplicações e estudos realizados por diversos autores, proporcionando um maior conhecimento sobre possíveis métodos a utilizar no desenrolar do trabalho.

No Capítulo 3 são apresentados os equipamentos utilizados durante esta dissertação, bem como descrito o ambiente criado para a validação dos resultados. Também neste capítulo é apresentado o processo de calibração de câmara, necessário à obtenção das coordenadas físicas do objeto de interesse, bem como do processo de seleção da cor a seguir. Estes processos aparecem descritos neste capítulo pois são dois módulos executados independente do módulo principal onde se encontra implementado o algoritmo de *tracking*.

O terceiro e ultimo capítulo descreve o módulo principal deste sistema, o algoritmo de *tracking*. Nesse sentido, é descrito todo o sistema desde a aquisição de imagem até à aplicação do processo de seguimento, bem como a obtenção das coordenadas físicas a enviar ao drone.

Capítulo 2

Estado da Arte

Neste capítulo pretende-se contextualizar o tema desta dissertação. O objetivo passa por realizar um levantamento teórico de tecnologias e aplicações já existentes de sistemas de visão computacional que implementem sistemas de *tracking* em Drones.

Para obter todos os dados necessários para este capítulo, recorreu-se a um conjunto de artigos com abordagem aos temas de deteção e *tracking* de objetos ou regiões de interesse. Com a variedade de informação existente é possível identificar, pelas diversas propostas e opiniões, um conjunto de etapas que devem ser adotadas com vista ao tema principal desta dissertação .

Com a globalização dos sistemas UAVs foi natural o aparecimento de sistemas úteis para a sociedade. Quando surgiram os primeiros UAVs, o intuito passava por auxiliar as forças militares [12]. Mas, com o passar do tempo, esta tecnologia globalizou-se passando a estar disponível a qualquer pessoa e foram surgindo diversos projetos e produtos comerciais. Neste capítulo serão ilustrados alguns destes casos.

2.1 Floresta

Em 2016, durante o evento IEEE ICUAS, um conjunto de autores apresentaram no seu trabalho a deteção de fogos florestais com recurso a imagens aéreas capturadas por drones [5]. Como é do conhecimento geral, um dos grandes flagelos mundiais são os fogos florestais. Com isto, verifica-se que os ecossistemas são constantemente colocados em causa, bem como a segurança das populações, sendo uma mais valia detetar estas ocorrências e atuar rapidamente.

A maioria dos métodos de deteção de incêndios é feito por pessoas onde a sua segurança pode ser colocada em causa, para além do atraso inerente em informar o sucedido. Com isto, um sistema automático de deteção torna-se uma vantagem, pois não coloca o ser humano em perigo, tendo ainda a vantagem da comunicação ser feita de forma mais rápida. Métodos de

deteção usando satélites, equipamentos terrestres ou aéreos (UAVs) tornam o combate aos fogos florestais mais eficazes e céleres.

Com o passar do tempo, o uso dos UAVs tornou-se importante nesta área pois com a possibilidade de incluir equipamentos externos permitiu recolher outro tipo de informações. O exemplo disso são as câmaras, com as quais foi possível aplicar técnicas baseadas em visão e assim obter dados, em tempo real, imprescindíveis para detetar fogos florestais.

Chi Yuan, Zhixiang Liu e Youmin Zhang propuseram um sistema de deteção de fogos florestais baseado na cor e nas características do movimento das áreas coloridas [5]. Quando potenciadas em conjunto, apresentam um nível muito elevado de fiabilidade no que respeita à sua deteção, como se pode comprovar pela Figura 2.1 que representa o fluxograma do algoritmo proposto.

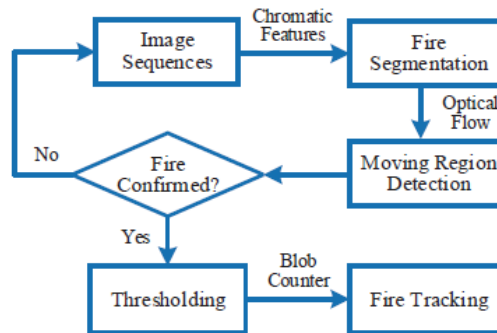


Figura 2.1: Fluxograma do algoritmo proposto em [5].

A análise da cor é uma das técnicas mais usadas na deteção de possíveis fogos florestais usando os sistemas de visão. Isto porque a cor da chama é conhecida podendo variar entre o vermelho e o amarelo. Perante tal situação, a deteção do fogo pode ser feita através da análise da cor, recorrendo aos espaços de cor RGB ou HSV. No entanto, a cor por si só pode não ser significado de que o incêndio se encontra ativo, pelo que poderá ser necessário recorrer a outros tipos de deteção. Neste caso, a proposta passa pela análise de possíveis regiões em movimento. Este método pode-se tornar interessante pois quando um fogo florestal está ativo, este tende em movimentar-se em consequência do vento. Com esta informação, o processo de deteção torna-se eficaz, pois em caso de regiões com cores semelhantes, mas onde uma se encontra estacionária, esta é descartada. Ainda assim, todos estes dados podem não ser suficiente, podendo ser necessário formas adicionais [5].

Experimentalmente, os autores deste sistema dividiram-no em duas partes. A primeira, analisando imagens já existentes, e uma segunda em contexto real. Nesta situação, utilizaram uma câmara colocada num drone e programaram este para percorrer um percurso pré-definido

até encontrar uma situação real de incêndio. A Figura 2.2 representa o sistema usado para a realização do sistema proposto.

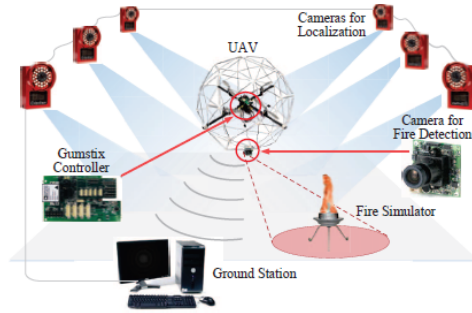


Figura 2.2: *Layout* experimental para monitorização e deteção de incêndios baseado em UAVs [5].

Na Figura 2.3 pode-se visualizar os testes realizados com imagens já existentes. Com este conjunto de tarefas conseguiram comprovar o funcionamento do seu método numa primeira fase.

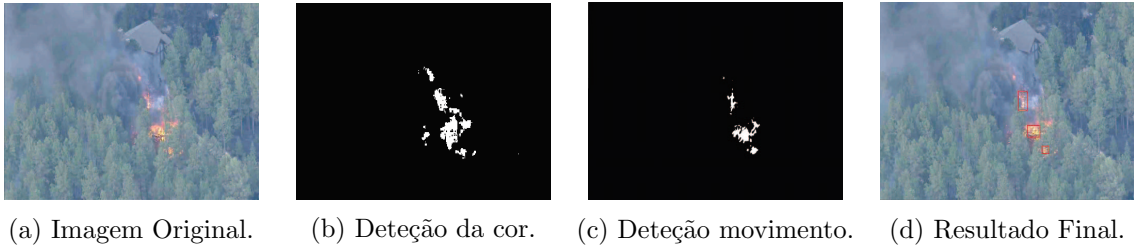


Figura 2.3: Evolução do processamento para a primeira parte [5].

A Figura 2.4 valida os testes realizados anteriormente. Se as imagens usadas fossem num contexto real de incêndio, era necessário validar o processo na prática. Na segunda parte, os testes foram realizados em laboratório simulando um incêndio e com um drone foi testado o algoritmo elaborado, sendo aprovado o seu funcionamento.

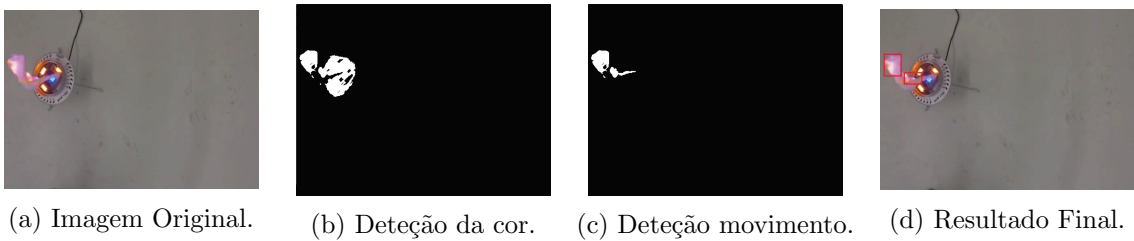


Figura 2.4: Evolução do processamento para a segunda parte [5].

2.2 Monitorização de pessoas

A dificuldade de conhecer a localização de possíveis sobreviventes soterrados em catástrofes é enorme. Concretamente, no caso das avalanches, existe a possibilidade de os encontrar nos minutos seguintes ao acontecimento. Em suma, a rápida assistência pode salvar vidas sendo para isso necessário uma forma eficaz de deteção das vitimas.

Com o intuito de melhorar a forma e método de atuar na procura de pessoas desaparecidas em avalanches, um conjunto de investigadores procuraram encontrar uma solução viável e eficaz. Com isto, apresentaram um projeto que visa a procura de pessoas em avalanches com recurso à tecnologia *Fourth Generation (4G) Long Term Evolution (LTE)* [13], sendo aplicado o sistema num drone.

A primeira fase realizada pelos autores de [13] foi testar o tipo de sinal 4G LTE recebido por um equipamento soterrado e não soterrado na neve, com o intuito de comparar os dois casos. Com este estudo foi possível perceber o comportamento do equipamento sem que um meio físico o ative. Concluíram que era necessário desenvolver uma aplicação que fizesse o processo de transmissão de uma forma automática. A aplicação têm como objetivo reunir um conjunto de dados do 4G LTE que serão enviados para os dispositivos acima descritos. Com os dados trocados, intensidade de sinal, é possível determinar as distâncias e consequentemente a localização exata dos equipamentos.

Na Figura 2.5 é possível verificar a curva característica da intensidade de sinal em relação à profundidade do equipamento transportado por uma vitima de avalanche.

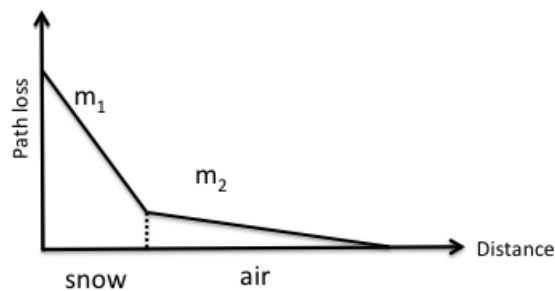


Figura 2.5: Curva de intensidade de sinal em relação à profundidade [13].

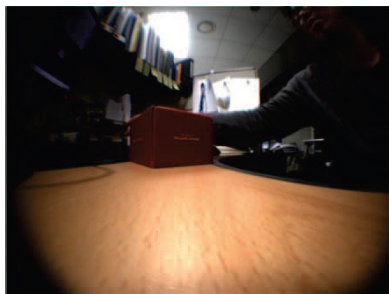
No entanto, o estudo realizado por estes autores não foi aprofundado o suficiente em relação à informação existente. Ou seja, a investigação realizada sobre este tema é escassa sendo por isso de salientar que este trabalho permitiu perceber que existe a possibilidade de salvar vitimas de avalanches de forma eficiente e eficaz, utilizando os sistemas de análise de sinais provenientes dos próprios dispositivos eletrónicos.

2.3 Sistemas de *Tracking*

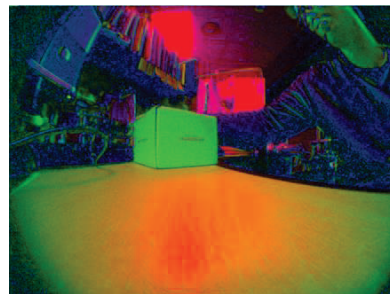
Os sistemas de *tracking* e detecção podem ser aplicados em diversas situações. Quando associado aos UAVs permitem obter sistemas robustos, capazes de realizar tarefas difíceis ao ser humano em tempo real.

Um exemplo deste tipo de sistemas é a aterragem de drones em veículos em movimento [7]. De um modo geral, o que os autores deste trabalho sugerem é detetar o veículo através de um marcador colorido. Posteriormente, para aterrar o equipamento, com os dados adquiridos pelo sistema *tracking*, aplicam sistemas de controlo por forma a garantir a aterragem.

De forma a conseguir uma aterragem do UAV num veículo em andamento, optaram por colocar um objeto colorido no veículo em causa. Com este, o método de detecção tornar-se-ia mais simples pois aplicando um processo de segmentação de cor à imagem adquirida, o alvo sobressai sobre os restantes objetos no ambiente. Para alcançar uma segmentação mais eficaz, recorrem à representação de cor no espaço HSV, pois com esta é possível obter intervalos de dados (tonalidade ou saturação da cor) capazes de atuarem em qualquer ambiente. Na Figura 2.6 é possível verificar a imagem e a sua representação em HSV utilizada pelos autores, na obtenção do algoritmo de segmentação.



(a) Imagem RGB.



(b) Imagem HSV.

Figura 2.6: Imagem teste para o processo de segmentação de cor [6].

Com a representação da imagem no espaço HSV, obtêm-se a região de interesse, sendo que neste exemplo, usam o vermelho. Para tal, aplicaram uma binarização onde os parâmetros usados são valores extremos do modelo HSV para a cor em causa. Usaram ainda um conjunto de processos de filtragem com vista a obter o objeto sem ruído, como se pode verificar através da Figura 2.7, onde do lado esquerdo é possível ver a binarização pelo intervalos de valores HSV e na direita o processo de filtragem da mesma.

Com o processo de segmentação concluído, iniciaram o algoritmo de controlo à aterragem da aeronave. Todo este processo é determinado com recurso a um conjunto de formulas, usando os dados obtidos com a segmentação de imagem. Em suma, a primeira etapa passa por determinar o erro posicional com base nas coordenadas cartesianas permitindo, ao usar



Figura 2.7: Binarização pelo modelo HSV [6].

as posições x e y da imagem, representar a posição final. Finalmente, é necessário determinar a trajetória de aterragem e, para este parâmetro, o momento do alvo influencia, pois em caso deste estar estático, o amortecimento depende unicamente da velocidade do alvo, enquanto que em caso de movimento, a velocidade reduz o amortecimento. Além disso foi necessário obter mais alguns dados de controle como orientação e outras variáveis importantes à conclusão do trabalho.

Com tudo isto, surgiu um sistema de aterragem de UAVs em veículos em movimento como se comprova com o teste realizado pelos autores e apresentado na Figura 2.8.



Figura 2.8: Teste à aplicação desenvolvida [6].

O outro exemplo de *tracking* está relacionado com os pequenos UAVs. Com o desenvolvimento destas aeronaves, nomeadamente no que se refere à redução de custo e melhoria da capacidade de manuseamento, estes equipamentos tornaram-se bastante populares gerando algumas preocupações [1]. Desde logo a privacidade com a potencial captura de imagens sem autorização prévia. Para tentar minimizar a situação, surgiu uma proposta de solução que consiste na deteção e seguimento destes equipamentos. Existem vários métodos capazes de o fazer como:

- a) *Audio Detection*
- b) *Video Detection*
- c) *Thermal Detection*

d) *RADAR Detection*

e) *Radio Frequency Detection*

Esta proposta recorre ao mesmo método da anterior [7]. O processamento de imagem é feito com recurso ao modelo de cor HSV. Depois da aquisição de imagem, esta é convertida para HSV e obtida a máscara para o objeto pretendido (Figura 2.9b). Com esta máscara o utilizador conhece um conjunto de dados sobre o objeto, entre eles, as suas coordenadas na imagem, sendo possível verificar na Figura 2.9a as mesmas alocadas na imagem original e ainda o deslocamento do drone, onde a diferença entre *frames* permite verificar o seu rasto (Figura 2.9c).



(a) Imagem RGB em processo de (b) Máscara do UAV com base (c) Diferença entre *frames* após *tracking*.
no método HSV. determinar a máscara.

Figura 2.9: Imagem teste para o processo segmentação de cor [6].

Além de todo o processo referido, os autores também recorrem a algoritmos avançados para reconhecimento de objetos como o SURF. Este é usado para detetar e separar os drones dos restantes objetos. Em concreto, se for detetado movimento, o algoritmo é aplicado nessa zona para identificar o objeto, sendo que a distância do mesmo pode influenciar a precisão do processo de seguimento. Na Figura 2.10 pode-se verificar a aplicação deste algoritmo na identificação do objeto em causa - drones.

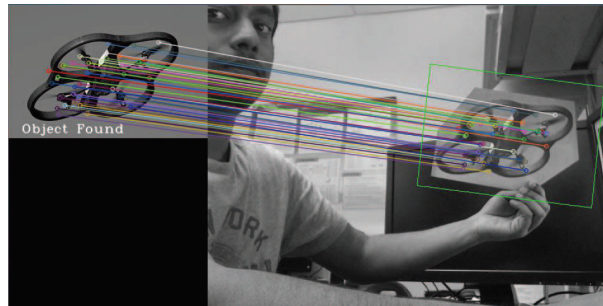


Figura 2.10: Aplicação do algoritmo SURF [6].

Capítulo 3

Sistema de Visão

Este capítulo tem como objetivo dar a conhecer o ambiente usado para testar e obter os dados necessários ao desenvolvimento desta dissertação. Além disso, serve também para apresentar alguns módulos independentes do módulo principal onde se encontra implementado o sistema de *tracking*: a calibração de câmara, necessária para a obtenção da posição física do objeto de interesse, e ainda a seleção, por parte do utilizador, da cor do objeto que pretende seguir.

3.1 *Hardware* e Sistema de testes

Como referido anteriormente, o objetivo deste trabalho passa pelo desenvolvimento de um sistema de visão computacional que implemente um sistema *tracking* e que permita ser incorporado num drone comercial. Para que seja possível a sua inclusão num drone comercial, normalmente plataformas fechadas, é necessário desenvolvê-lo num periférico independente deste. Assim permite também a sua utilização em qualquer tipo de equipamento.

Para desenvolver todo o *software* necessário à execução desta dissertação, utilizou-se um micro-computador, mais precisamente um Raspberry Pi (Figura 3.1). Este equipamento apresenta a vantagem de ser pequeno e leve por forma a ser suportado pelo drone. Outra vantagem é ser possível implementar os algoritmos necessários em linguagens de programação de alto nível, como por exemplo em python.

Outro equipamento utilizado foi uma câmara digital: a PiCamara (Figura 3.2), cujas suas características podem ser visualizadas no Anexo 6. Como o objetivo do presente trabalho refere, pretende-se detetar e seguir um determinado objeto de interesse através de um drone e imagem digital, daí a importância deste componente.

Ao utilizar um drone como equipamento seguidor e como a maioria destes apresentam câmaras incorporadas, poderia-se ter usado a este equipamento. No entanto, isso não se

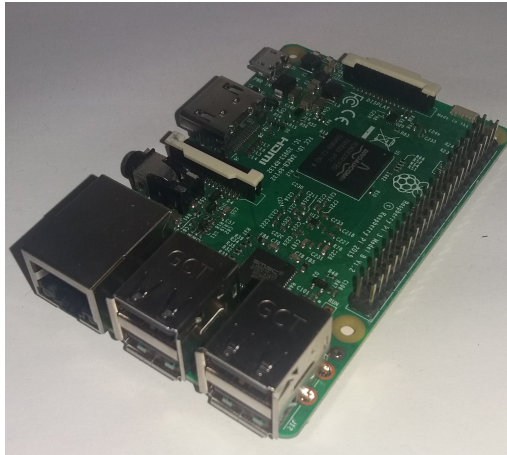


Figura 3.1: Raspberry Pi.

verificou pois como se pretendia processamento em tempo real não é compatível tal uso. Esta incompatibilidade deve-se ao facto de serem equipamento separados, drone e microcomputador, o que implica uma troca de informação, por wifi ou outro método, e consequentemente um tempo para tal troca. Ao utilizar uma câmara ligada diretamente ao Raspberry Pi minimiza-se tal fator, tornando o sistema mais próximo do real.

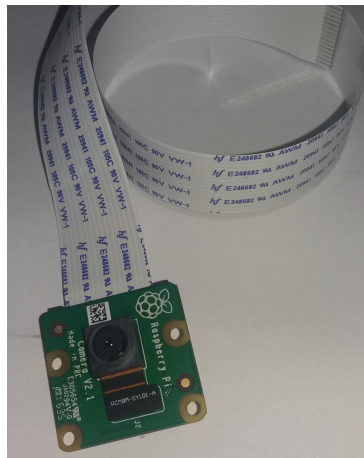


Figura 3.2: Picamara.

Para testar os algoritmos desenvolvidos foi simulada uma plataforma para o efeito. Esta têm como intuito a validação dos resultados com vista à posterior implementação do sistema no drone.

Ao gerar tal plataforma, teve-se em conta a aplicação final do sistema. Como prova de conceito será o seguimento objetos coloridos, supõe-se que estes estejam num ambiente externo com luz natural e todas as condicionantes envolventes, como a mudança de luz num dia de sol, entre outras.

Deste modo, foi colocada uma câmara a uma determinada altura, no caso cerca de 2m, para simular a altitude do drone. Salientar que a mesma, para efeitos de teste, foi colocada num local estático. No que respeita ao meio envolvente, os testes são realizados em meio externo, com luz natural. Os objetos de simulação apresentam duas cores e dois formatos com diversas dimensões. Isto permite validar a primeira parte do trabalho, que visa a seleção a cor de interesse seguir e, também, o processo de *tracking*. Na Figura 3.3 pode-se verificar o ambiente criado para validar os resultados.



Figura 3.3: Ambiente de testes criado para validação de resultados.

e referir que todos os dados e validações obtidas neste documento tem como base de testes a plataforma abordada nesta secção.

3.2 Calibração de câmara

A calibração de câmaras é um processo essencial para a conversão de coordenadas, ou seja, obter coordenadas físicas com base na posição numa imagem [9]. Este é um processo

que envolve a caracterização da câmara por parâmetros intrínsecos e extrínsecos. O primeiro relaciona as propriedades geométricas da câmara enquanto que o segundo relaciona a sua posição com um dado referencial.

Os parâmetros intrínsecos relacionam coordenadas numa imagem com base no referencial da câmara. Um exemplo utilizado na descrição da geometria da imagem é o modelo *pinhole*, Figura 3.4, que consiste na formação da imagem tendo em conta um pequeno orifício onde apenas o atravessa um raio de luz por cada ponto da imagem.

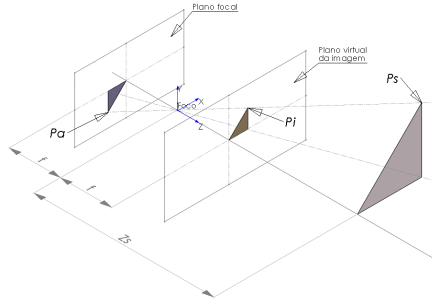


Figura 3.4: Modelo pinhole

Tendo em conta todas as condicionantes inerentes às câmaras digitais é possível obter um conjunto de Equações, 3.1 e 3.2, da qual é possível obter a relação de coordenadas da imagem em relação ao mundo físico.

$$\mathbf{K} = \begin{bmatrix} \alpha_x & 0 & x_0 \\ 0 & \alpha_y & y_0 \end{bmatrix} \quad (3.1)$$

$$Z_s \times \begin{bmatrix} X_{pix} \\ Y_{pix} \end{bmatrix} = \mathbf{K} \times \begin{bmatrix} X_s \\ Y_s \\ Z_s \end{bmatrix} \quad (3.2)$$

A obtenção dos parâmetros intrínsecos é realizada com recurso a um conjunto de imagens padrão. Estas imagens devem ter geometria conhecida como a base de xadrez.

A calibração da câmara utilizada nesta dissertação é realizada com recurso a um algoritmo já existente da biblioteca OpenCV, *cv::calibrateCamera*, onde foram usadas diversas imagens padrão capturadas pela mesma, como a da Figura 3.5a.

Após a execução do algoritmo é possível obter a matriz intrínseca da câmara utilizada, expressão 3.3, bem como outros parâmetros tais como o coeficiente de distorção. Este último é aplicado nas imagens pois as lentes, quando recebem a luz para formar a imagem no sensor, introduzem alguma distorção sendo necessário aplicar parâmetros para corrigir a localização dos pontos. Estes dados têm como base o ambiente de testes e equipamentos abordados na



(a) Imagem padrão de xadrez.



(b) Detecção dos cantos do xadrez.

Figura 3.5: Processo de calibração de uma câmara.

secção 3.1.

$$\mathbf{K} = \begin{bmatrix} 1180,37 & 0 & 546,59 \\ 0 & 1198,91 & 424,24 \end{bmatrix} \quad (3.3)$$

Estes dados serão bastante úteis no desenrolar desta dissertação, pois a matriz intrínseca será muito importante para determinar a relação de coordenadas, visto que é necessário o conhecimento das coordenadas físicas do objeto a seguir para fornecer ao equipamento seguidor, neste caso o drone.

Relativamente aos parâmetros extrínsecos, estes permitem relacionar a localização da câmara (P_c) com o sistema de coordenadas físicas (P_w). Esta relação pode ser determinada analisando a Equação 3.4, onde R e T correspondem às matrizes de rotação e translação do mundo físico vista pela câmara.

$$P_c = R \times (P_w - T) \quad (3.4)$$

As expressões 3.5 e 3.6, representam os vetores de parâmetros extrínsecos, mais precisamente o vetor de rotação e translação da câmara em relação ao mundo, determinado pela biblioteca OpenCV de calibração de câmara.

$$rotation = \begin{bmatrix} 0,5128 \\ -0,6499 \\ 0,3225 \end{bmatrix} \quad (3.5)$$

$$translation = \begin{bmatrix} -3,4929 \\ -2,8269 \\ 16,0611 \end{bmatrix} \quad (3.6)$$

3.3 Seleção da cor de interesse

Detetar um objeto é a primeira fase num sistema de *tracking*. No caso desta dissertação, o método escolhido foi a deteção por cor. Como é conhecido, esta apresenta um conjunto de características que permitem, de uma forma mais simples e rápida, determinar o alvo pretendido. No entanto, este não pode ser unicamente o método de deteção devido à sua amplitude de trabalho. Ao procurar uma cor na imagem, esta pode aparecer em diversos locais sendo, à posteriori, importante aliar os outros métodos como a forma, no intuito de centralizar ao objetivo.

Nsta Secção, pretende-se dar a conhecer o trabalho inicial para detetar um objeto e definir a cor de interesse. Este processo ocorre independente do algoritmo de *tracking*, ou seja, são programas diferentes com o intuito de maximizar o processo principal, seguir objetos de interesse.

De seguida, irão ser apresentados os processos de deteção de cor e, ao mesmo tempo, os testes de validação dos mesmos. Estes processos consistem no método de *Floodfill*, método manual (*slider*) ou análise do histograma.

3.3.1 *Floodfill*

O método de *Floodfill* é um processo que consiste no enchimento de objetos. Ou seja, com base num determinado pixel, o processo vai analisar a sua vizinhança sendo a conectividade determinada pela proximidade da cor ou brilho.

O processo elaborado para esta dissertação segue um conjunto de tarefas até à obtenção dos dados necessários, que nesta primeira fase do trabalho consiste na gama de valores HSV a utilizar no programa principal, algoritmo de *tracking*. Para obter tais dados, o primeiro passo é dado pelo utilizador ao selecionar na imagem a cor do objeto a seguir. Ao clicar num ponto, o processo de *floodfill* é iniciado na imagem convertida para HSV. Após execução deste, é obtido os valores mínimos e máximos para os três parâmetros HSV para a cor selecionada. Estes são guardados num ficheiro de texto (.txt) por forma a transporta-los para o algoritmo de *tracking*.

Na Figura 3.6 é possível observar todo o processo referido. Neste caso, a cor escolhida foi o vermelho tendo sido obtido os seguintes valores:

- HSVmin = [208, 30, 116]
- HSVmax = [245, 127, 195]

Este processo automático tem a vantagem de se tornar mais rápido do que os seguintes, pois não está dependente de terceiros. Assim, o algoritmo desenvolvido para obter os

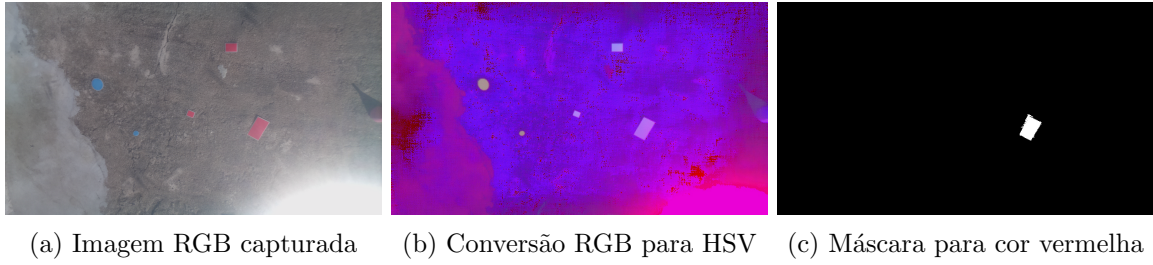


Figura 3.6: Escolha da cor a seguir por processo de *floodfill*

parâmetros de segmentação da cor de interesse, precisa de cerca de 60ms, após o utilizador clicar sobre a cor pretendida, para determinar o mínimo e máximo das três componentes do modelo HSV.

3.3.2 Seleção Manual (*Slider*)

Outro processo para obtenção da cor pretendida é o método manual. Isto significa, que o ajuste dos valores é feito manualmente, aumentando ou diminuindo os mesmos nas diferentes vertentes do modelo HSV. De todos é o menos aconselhável pois a precisão da gama é mais difícil de obter em relação ao antecedente.

O processo manual desenvolvido para esta dissertação é semelhante ao anterior, com a diferença a situar-se, como referido, na forma de obter os limites de HSV. Ou seja, após a aquisição de imagem (Figura 3.7a) e sua conversão para o modelo HSV (Figura 3.7b), o utilizador clica sobre a cor que pretende. Este clique permite obter uma referência sobre valor HSV pretendido, permitindo-o ajustar de forma mais rápida e eficaz. Após adicionar ou subtrair valores a cada item, mínimo e máximo para as três componentes de HSV com recurso a sliders na imagem, é alcançada a máscara aproximada do objeto colorido (Figura 3.7c) e consequentemente os valores a transportar para o programa principal através do ficheiro texto.

A Figura 3.7 representa o processo manual desenvolvido. Para as mesmas condições do processo anterior, foram obtidos os seguintes resultados:

- HSVmin = [235, 94, 145]
- HSVmax = [249, 146, 181]

Nota ainda para o tempo de execução deste algoritmo. Este é variável pois está dependente do tempo que o utilizador leve para ajustar os valores de forma manual.

3.3.3 Histograma

Além dos sistemas referidos anteriormente, a análise por histograma pode ser outro método a usar para a calibração da cor de interesse. Este processo pode ser aliado ao anterior, onde

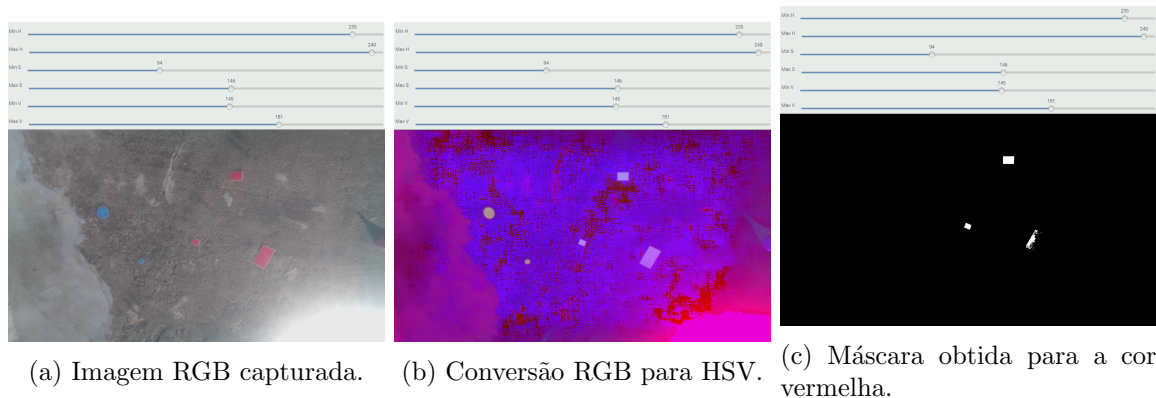


Figura 3.7: Escolha da cor a seguir por processo de seleção manual (*slider*).

os valores a retirar do histograma são obtidos em paralelo com a seleção manual. Na Figura 3.8 pode-se observar três histogramas diferentes para a Figura 3.7. O primeiro corresponde ao modelo RGB, o segundo ao modelo HSV sendo o ultimo correspondente à componente H do modelo anterior.

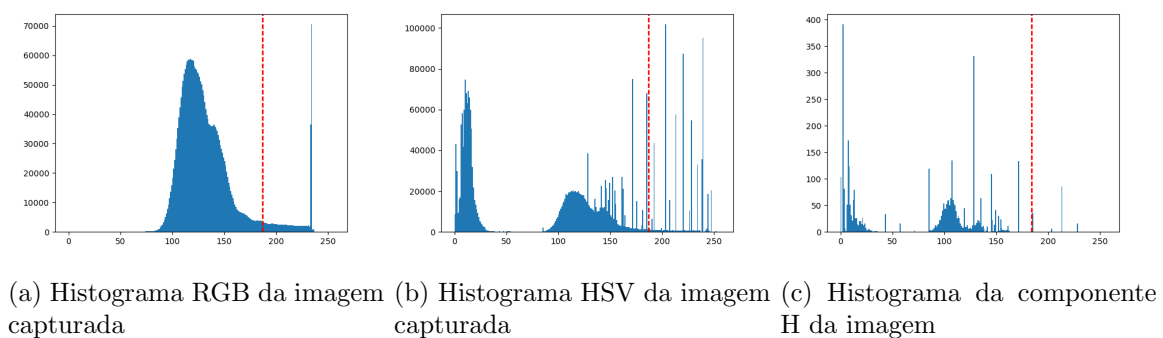


Figura 3.8: Escolha da cor a seguir por análise do Histograma

Capítulo 4

Sistema de *tracking*

Fazer *tracking* a um objeto significa que este vai ser seguido ou acompanhado [16, 15]. Este processo pode ser aplicado a um único objeto (*single-tracking*) ou vários objetos (*multi-tracking*), sendo que o resultado final deriva do primeiro. Com este capítulo pretende-se apresentar o algoritmo desenvolvido para fazer *tracking*. Deste modo, irão ser descritos os blocos resultantes no objetivo final.

4.1 Identificação dos objetos coloridos e possíveis alvos

Um dos aspetos principais e iniciais num algoritmo de *tracking* passa pela identificação do alvo a seguir. No entanto, este pode não ser o único a aparecer numa imagem. Pelo que poderá ser essencial isolá-lo.

Pretende-se explicar de seguida o processo referido para a identificação dos objetos coloridos, possivelmente alvos, com o intuito que aplicar, à posteriori, o processo de *tracking*.

O processo de identificação dos objetos tem como base a Secção 3.3. Como foi referido, a primeira rotina a ser executada é o processo de seleção de cor onde, como explicado, o objetivo passa por visar a cor do objeto a seguir. Com este processo é permitido obter valores máximos e mínimos do modelo HSV, os quais serão usados para isolar a cor pretendida.

Usar o espaço de cor HSV (Figura 4.1) para identificar e isolar cores é o mais usual [10]. Também é possível recorrer a outros modelos, tais como o espaço RGB, CMYK, YCbCr, entre outros. Este é espaço de cor mais usado pela sua amplitude, pois a segmentação usando os 3 canais, H, S e V, permite obter diversas variantes que vão desde as mais claras às mais escuras.

Desta forma obtêm-se a máscara dos objetos com a cor pré-definida. Isto é, toda a imagem capturada é convertida para o espaço de cor HSV e utilizando os parâmetros adquiridos na Secção 3.3, define-se a máscara onde a cor pretendida é isolada das restantes. Com este último

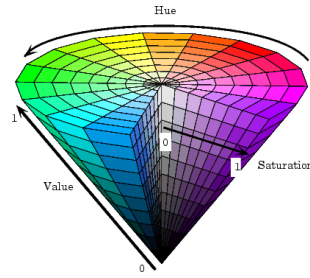


Figura 4.1: Espaço de cor HSV

dado pode dar-se por concluído o processo de segmentação da cor, sendo que a máscara obtida é a peça essencial ao restante trabalho de identificação do alvo.

Com os objetos isolados, o passo seguinte passa por identificar o objeto de interesse. Este processo pode-se realizar de duas formas. Na primeira, o utilizador seleciona o objeto com um clique sendo que na segunda, a escolha é feita de forma automática através de um algoritmo. Para ambos os casos, a máscara é submetida à análise das suas propriedades geométricas com o intuito de conhecer alguns dados específicos. Com esta, obtêm-se características importantes que serão utilizadas, para além de identificar o objeto alvo, no processo de *tracking* do mesmo.

As propriedades geométricas que se podem extrair de um objeto são imensas. Como por exemplo, podemos considerar a área, perímetro, solidez, fator de forma, coordenada central (centroid) na imagem, entre muitos outros que caracterizam o objeto. Estes parâmetros inicialmente têm como objetivo isolar os objetos com determinados atributos, pois numa imagem pode aparecer mais do que um com a mesma cor e usando vários é possível isolar os objetos pretendidos pela sua área ou solidez. À posteriori, quando o alvo está definido, a coordenada central (centróid) do objeto é usada para o *tracking* e posteriormente no seu seguimento.

Como referido anteriormente, a identificação do objeto pode ser feita de duas formas: pelo utilizador ou por *software*. No primeiro caso, o utilizador tem apenas que clicar no objeto pretendido para o determinar como alvo. Quando a escolha é realizada pelo *software*, é este que determina o objeto de interesse a seguir usando como referencia, por exemplo, a área deste ou qualquer outra propriedade configurada previamente e que depende da aplicação.

Nas Figuras 4.2 e 4.3 é possível observar o resultado obtido com o algoritmo desenvolvido. No primeiro, *tracking* por objeto de maior área, observa-se as várias fases do processo, que passa pela deteção e marcação de algum objeto existente (Figura 4.2a) até à deteção de novo com as mesma características (Figura 4.2c).

O algoritmo processa as imagens até encontrar um objeto de cor vermelha. Se só encontrar um, marca-o. Como é único, tem a maior área e o ciclo é repetido (Figura 4.2a). Se entretanto surgir um novo na imagem, o algoritmo verifica qual dos objetos encontrados é maior (Figura

4.2b). Após essa análise, se o objeto inicialmente definido como maior continuar a sê-lo, então continua com a marcação de objeto a seguir. Caso contrário, esse rótulo passa para o novo objeto encontrado (Figura 4.2c).



Figura 4.2: Escolha do objeto com maior área a seguir.

No caso da escolha do objeto a seguir ser feita pelo utilizador, o algoritmo vai processando sucessivamente imagens até encontrar um objeto com a cor pretendida. Assim que o encontra, o utilizador é alertado com uma nova janela na qual deve selecionar o objeto alvo. Isto é feito independentemente de ter um ou mais objetos detetados na mesma imagem.

De salientar que o algoritmo não avança na sua execução enquanto o utilizador não selecionar o objeto de interesse na imagem. Ou seja, o programa só determina se é ou não objeto a seguir ou a distância de seguimento, se o utilizador clicar num ponto na imagem, supostamente contendo o objeto de interesse. Se o objeto tiver desaparecido, o programa volta a repetir o processo até encontrar possíveis objetos interesse.

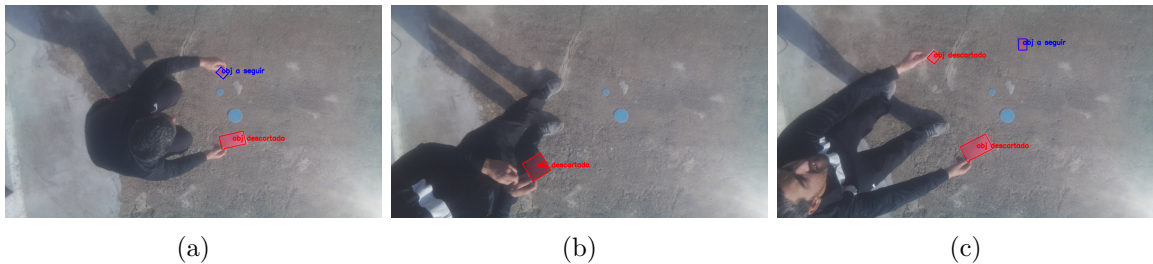


Figura 4.3: Seleção do objeto alvo pelo utilizador.

4.2 *Threshold de tracking*

O *tracking* de objetos tem como base a avaliação de duas sequências temporais, a análise das coordenadas no instante t e $t + 1$. De acordo com os dados obtidos é fácil perceber se o objeto está ou não a ser seguido. Nesta dissertação, o modo de *tracking* implementado tem como base o referido: a análise sequencial de imagens. Contudo, é necessário definir a zona de

tracking, isto é, a distância ou *threshold* que valide se o objeto está ou não a ser seguido. Se a diferença posicional do objeto de interesse for inferior ao *threshold*, significa que é o objeto pretendido. Caso contrário, é sinal que o mesmo foi perdido ou não é o alvo. Na Secção 4.3 será explicado mais em pormenor o funcionamento do processo de *tracking* desenvolvido.

O valor de *threshold* depende do objeto de interesse a seguir. Por exemplo, o valor de *tracking* para seguir uma pessoa a caminhar não deve ser o mesmo que para um carro, pois a velocidade de movimento é diferente nos dois casos. Assim sendo, é importante adequar o valor com o situação em causa.

A primeira tarefa passa por determinar qual a relação entre a imagem e o mundo físico (Z_s). Este pode ser determinado tendo como base a formação da imagem pelo modelo de pinhole. Analisando a Figura 4.4 e a Equação 4.1, é possível determinar esta dimensão sabendo um conjunto de dados, tais como a distância focal da câmara (df), altura a que a mesma se encontra do plano de interesse (h) e a densidade de pixels da câmara (Dp). Com esta informação é possível determinar a largura real vista pela câmara que posteriormente será utilizado no cálculo do *threshold*.

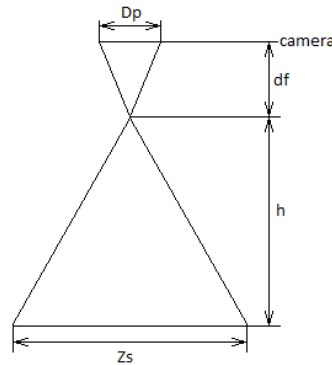


Figura 4.4: Diagrama de calculo do *Threshold* de *tracking*.

$$Z_s = \frac{Dp \times h}{df}$$

Dp - distância em mm da largura da imagem; (4.1)

h - altura da câmara;

df - distância focal da câmara;

Os dados necessários à resolução da Equação anterior são facilmente obtidos através da *datasheet* da câmara ou por calibração da mesma. No sistema desenvolvido nesta dissertação, o tamanho do pixel da câmara é de $1,2\mu m$, a distância focal de $3,04mm$ e que está colocada a $2,1m$ de altura, de acordo com o *setup* de testes descrito em 3.1. Tudo isto, recorrendo a

imagens com resolução de 720p, ou seja, 1280×720 pixels.

1º passo - Cálculo da variável D_p : este dado é determinado com base na largura da imagem e no tamanho do pixel da mesma. Equivale dizer que:

para largura da imagem: $D_p = 0,0012 \times 1280 = 1,536\text{mm}$

para altura da imagem: $D_p = 0,0012 \times 720 = 0,864\text{mm}$

Pelo que, para uma resolução de 1280×720 , a câmara gera uma imagem de $1,536 \times 0,864\text{mm}$, ou seja, uma área aproximada de $1,33\text{mm}^2$.

2º passo - Cálculo de Z_s : conhecendo a largura da imagem na câmara, o passo seguinte é determinar a amplitude real da câmara. Assim, ao aplicar a Equação 4.1 é possível obter os seguintes valores:

para largura da imagem: $Z_s = \frac{1,536 \times 2100}{3,04} = 1061\text{mm} \approx 1,1\text{m}$

para altura da imagem: $Z_s = \frac{0,864 \times 2100}{3,04} = 597\text{mm} \approx 0,6\text{m}$

Deste modo, com a câmara colocada a 2,1m do solo, a amplitude real que esta alcança é de aproximadamente $1,1\text{m} \times 0,6\text{m}$, ou seja, uma área útil de $0,66\text{m}^2$.

Por fim, após a obtenção dos dados anteriores é possível determinar qual o *threshold* a usar. Admitindo que o objeto de interesse se move no máximo 50mm entre imagens, pode-se verificar que os 1280 pixels correspondem a 1,1m na realidade. Logo os 50mm corresponderão a uma determinada quantidade de pixels. Neste caso, usa-se como exemplo os 50mm como teste pois os objetos usados são de pequena dimensão, e como a câmara se encontra fixa, não é possível utilizar grandes movimentos sob pena dos objetos saírem do alcance da mesma e, inviabilizar os testes ao sistema. Num contexto final, poderia-se usar 1,5m para seguir pessoas visto ser uma possível velocidade de caminhada ou no caso de seguir automóveis usar-se 14m. Neste caso, os 14m refere-se a uma velocidade média de cerca dos 50km/h.

$$th = \frac{1280 \times 0,05}{1,1} \approx 58\text{pixels} \quad (4.2)$$

Em modo de conclusão, os passos referidos permitem idealizar o valor real de *threshold*, o que garante fiabilidade no momento de implementar o processo de *tracking*. Com os dados anteriores, conclui-se que este valor é de aproximadamente 58 pixels. Para que o objeto esteja a ser seguido, a diferença entre a localização atual com a anterior (entre imagens) não pode superar tal valor sob pena de sair da zona de *tracking*.

4.3 Processo de *Tracking*

O processo de *tracking* advém da Secção anterior. Não é possível seguir um objeto se este não for identificado para tal. Este processo é realizado com o intuito de associar objetos em imagens sequenciais ou vídeos. Todavia, nem sempre é fácil associar os objetos em *imagens* sequenciais. Isso pode dever-se a diversas situações, tais como uma movimentação mais rápida ou em casos mais complexos, uma mudança de direção. Em muitas situações, os próprios algoritmos já estão preparados com modelos que podem simular esses casos [15].

O seguimento de um objeto pode ser feito de diversas formas. Analisando uma região da imagem em torno de um determinado ponto ou através da medição de deslocamento. No primeiro, é delimitada uma região onde o objeto é seguido se estiver contido dentro da mesma. Supondo que os objetos 1 e 2 são o mesmo, mas distados de t segundos. Veja-se como exemplo a Figura 4.5, dois exemplo deste método. Na Figura 4.5a, verificar-se que o objeto é seguido pelo facto de o mesmo se encontrar dentro da região delimitada de *tracking* (quadrado tracejado). Enquanto que na Figura 4.5b observa-se o contrário. O objeto está deslocado da zona delimitada, logo não é definido como objeto a seguir.



(a) Objeto dentro da região de *tracking*.

(b) Objeto fora da região de *tracking*.

Figura 4.5: Exemplo do processo de análise de regiões.

No segundo caso, a distância entre posições nas imagens é outro método a referir. Este têm em conta um determinado ponto do objeto (por exemplo o centroide), que será analisado a cada iteração. Se a distância entre cada posição na imagem for menor que um determinado valor, então é o objeto que vai ser seguido.

De salientar que as zonas/regiões ou gamas de *tracking* respeitam um conjunto de regras à sua definição. Na Secção 4.2 é explicado a forma de obter o valor para o método de distâncias, ou seja, o valor para o qual o objeto deve ser considerado seguido ou não. Nesta dissertação, o algoritmo elaborado utiliza o último método.

Como é executado o processamento de imagem antes de iniciar o processo de *tracking*, são conhecidos as diversas características do objeto em causa, entre elas a coordenada central. Assim, é usado o *centroid* do objeto em ambas as imagens para determinar a distância

percorrida. Ao determinar-la, objeto alvo e outros que possam estar contidos na imagem, o algoritmo compara tais distância com o valor de *threshold*, determinado com base na descrição da Secção 4.2. Se alguma for inferior ao valor pré-definido de *tracking*, é sinal da existência de um objeto a seguir sendo a ordem de seguimento dada. Caso contrário, o algoritmo continua a procura do objeto de interesse.

Na tabela que se segue (Tabela 4.1), é possível observar alguns valores de *tracking*, sendo que no Anexo 7.1 pode-se encontrar uma lista mais completa para esta experiência. De salientar ainda que o teste para a obtenção destes valores corresponde à execução do algoritmo de escolha por maior objeto. A fase experimental do algoritmo de teste pretende demonstrar, para as diferentes posições do objeto, se este está ou não em zona de *tracking*. Este valor é determinado de acordo com a representação da Secção 4.2. Assim, o valor a verificar cifra-se em aproximadamente 84 pixels, tendo em conta as seguintes características:

- Distância da câmara ao ponto mais afastado $\rightarrow 2,1\text{m}$
- Distância de movimento do objeto $\rightarrow 50\text{mm/s}$
- Pixel size da câmara $\rightarrow 1,2\mu\text{m}$
- Distância focal da câmara $\rightarrow 3,04\text{mm}$
- Dimensão da imagem $\rightarrow 1280 \times 720$

O teste aqui apresentado, ao algoritmo de escolha por área, consiste em seguir o maior objeto vermelho (Figura 4.6). Foi criando uma ambiente teste, onde foi utilizado mais do que um elemento da mesma cor. Isto é, foram usadas duas cores (azul e vermelho), sendo que para cada uma existiam dois ou mais objetos com tamanhos diferentes, como se pode ver na Figura 4.6a. O ambiente referido tem como condição principal a localização da câmara, sendo que para o caso esta se encontra fixa a cerca de 2 metros de altura.

Como se pode verificar na Figura 4.6, existem dois objetos vermelhos pelo que é necessário identificar o objeto de interesse. O algoritmo analisa os dois e verifica qual deles é o maior, marcando-o como objeto a seguir.

Com os dados representados é possível verificar que nenhum das coordenadas ultrapassa o valor de *threshold*. No máximo, o objeto movimentou-se 52 pixels (item 2). No entanto, nas coordenadas onde a distância tem valor 0 (item 1) é sinal de que o objeto se movimentou, ao ponto de o algoritmo o descartar. Procurando assim um novo objeto de interesse.

Referir que nos Anexo 7.1 e 8 pode-se encontrar outros dados referentes ao ao tipo de seleção do objeto de interesse a seguir, escolha por maior área e pelo utilizador, respetivamente.



Figura 4.6: Exemplo da detecção de dois objetos com a mesma cor

Tabela 4.1: Distância entre as posições do objeto a seguir (posição atual e posição anterior)

item	centróid do objeto	centróid da posição anterior	distância de tracking (pixels)
1	(532, 149)	(532, 149)	0,0
2	(529, 97)	(532, 149)	52,0865
3	(533, 80)	(529, 97)	10,198
4	(523, 78)	(533, 80)	18,439
5	(505, 74)	(523, 78)	10,198
6	(484, 71)	(505, 74)	21,213
7	(460, 51)	(484, 71)	31,241

4.4 Controlo

No caso do sistema de visão ter encontrado o objeto de interesse pode-se então transmitir o deslocamento necessário ao drone. Esta Secção tem como objetivo descrever a primeira parte do controlo do mesmo. Isto é, a distância necessária à sua movimentação por forma a seguir o objeto de interesse.

Assim sendo, o primeiro passo é determinar a distância percorrida pelo objeto na imagem. Esta é obtida pela análise sequencial entre duas imagens. Será ainda necessário definir o ponto de referência a usar para o processo, ou seja, o local para onde o objeto deve ser colocado após cada iteração de movimento. De um modo geral, este ponto poder ser qualquer um dentro da imagem, dependendo da aplicação, sendo o mais comum a utilização do centro da imagem de modo a que o drone fique centrado com o objeto (no caso da câmara se encontrar na vertical). A Figura 4.7 serve de auxílio à explicação deste processo. De salientar que a imagem tem como dimensões 1080x720pixels.

Como referido anteriormente, o posicionamento do objeto na imagem pode ser qualquer um. No entanto, o mais usual e que foi utilizado no desenrolar desta dissertação, foi o ponto central da imagem. Ao utilizar este ponto para reposicionar o objeto na imagem, facilita bastante o desenvolvimento do sistema de controlo do drone. Assim sendo, o ponto de referência determinado foi $P0 = (640,360)$ de acordo com a dimensão das imagens processadas,

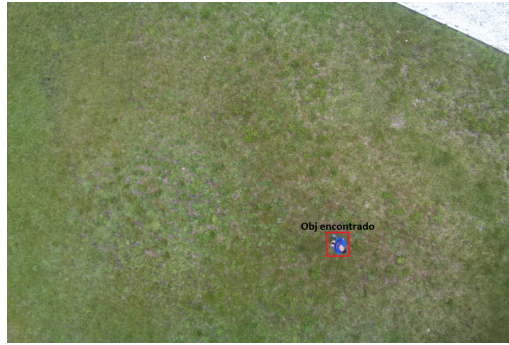


Figura 4.7: Exemplo do posicionamento do objeto de interesse a seguir.

1280x720.

A distância de movimento do objeto na imagem é determinada pela diferença posições. É estimada a diferença entre a posição do objeto com o ponto de referência, como é demonstrado de seguida.

- $d = P1 - P0 = (532, 149) - (640, 360) = (-108, -211)$

Fazendo a diferença de coordenadas, verifica-se que o objeto se movimentou 108 pixels na horizontal (eixo X) e 211 pixels na vertical (eixo Y), não tendo em conta a rotação.

- $d = \sqrt{X^2 + Y^2} = \sqrt{(-108)^2 + (-211)^2} = 237,034$

Com este cálculo obtém-se, então, a distância **d**. Ou seja, a distância percorrida pelo objeto na imagem.

Na Tabela 4.2 está descrito um excerto dos resultados obtidos para o seguimento do objeto na imagem, de acordo com o ambiente de testes criado. Nos Anexos 7.1 e 8 podem-se encontrar tabelas de dados mais completas para o algoritmo de escolha por maior área e pelo utilizador, respetivamente.

Tabela 4.2: Resultado da distância entre o objeto e o centro da imagem.

item	centróid do objeto	Coordenada referência	distância de movimento(pixels)
1	(532, 149)	(640, 360)	237.034
2	(529, 97)		285.464
3	(533, 80)		299.784
4	(523, 78)		305.308
5	(505, 74)		316.261
6	(484, 71)		328.416
7	(460, 51)		357.604

Capítulo 5

Conclusões e Trabalho futuro

Com a realização desta dissertação pretendia-se o desenvolvimento de um sistema de *tracking* autónomo que pudesse ser incorporado num drone comercial. Para atingir o objetivo proposto, foi necessário ultrapassar algumas fases importantes.

A primeira consistiu no estudo de sistemas já desenvolvidos o que permitiu obter conhecimentos importantes e possíveis métodos a aplicar no desenrolar desta dissertação.

No que refere ao trabalho prático, a primeira tarefa a realizar foi a seleção da cor de interesse a seguir. Este é um dos processos importantes pois é uma forma possível de identificar um alvo de interesse. O intuito desta fase do trabalho é a obtenção dos valores do modelo HSV, para a cor pretendida, a utilizar na segmentação do algoritmo de *tracking*. A obtenção deste dados pode ser realizada de uma forma automática ou manual. No primeiro caso é o próprio, recorrendo rotinas como *floodfill*, que devolve os valores necessários após o utilizador escolher a cor. No segundo caso, este processo é realizado pelo próprio utilizador, ajustando os valores HSV manualmente. Com estas formas de seleção de cor, pode-se afirmar que o método automático é mais viável visto ser mais rápido pelo facto de não depender do tempo de ajuste manual. Salientar, ainda, a obtenção de uma margem maior de valores HSV em relação ao outro método referido.

O o algoritmo de *tracking* é o ultimo e derradeiro processo desenvolvido. A este está associado um conjunto de ações como a identificação do objeto de interesse ou a obtenção dos dados a enviar ao equipamento seguidor. A identificação do objeto de interesse tem como objetivo dizer ao sistema qual é o objeto a seguir. Esta identificação pode ser realizada de várias formas, como o utilizador dizer ao algoritmo qual pretende seguir ou sendo o próprio algoritmo a fazer-lo. Neste último caso, foi dada a ordem de o algoritmo escolher o objeto de interesse de maior área.

Com o objeto de interesse a seguir definido, foi aplicado a este o processo de *tracking*. Este processo valida se o objeto escolhido está ou não a ser seguido de acordo com a sua

movimentação. Para verificar a referida informação foi determinado um *threshold* que verifica se o objeto está ou não a ser seguido. Pelos testes realizados, foi possível verificar que o objeto é dado como sendo o a seguir se não se não aumentar a velocidade pré-definida. Neste caso, o objeto era desacreditado sendo verificado, novamente, a sua posição e aplicado o processo de *tracking*.

Por fim, com o objeto bem identificado como objeto de interesse a seguir era necessário a obtenção dos dados físicos a ceder ao equipamento seguidor. Neste caso, apenas foi determinado o valor de movimentação do drone de acordo com a imagem.

5.1 Trabalho Futuro

O resultado final, apesar de um dos objetivo principais ter sido atingido, pode ainda ser bastante melhorado. Ou seja, como trabalho futuro sugere-se melhorar os seguintes pontos:

- Conclusão do sistema de calibração automática utilizando a análise por histograma;
- Completar e melhorar o modo de comunicação entre equipamentos;
- Interligar os três sistemas: Drone, sistema de *Tracking* (unidade de processamento) e utilizador (um computador, por exemplo).

Capítulo 6

Anexo A

6.0.1 Caraterísticas Pi Câmera [11]

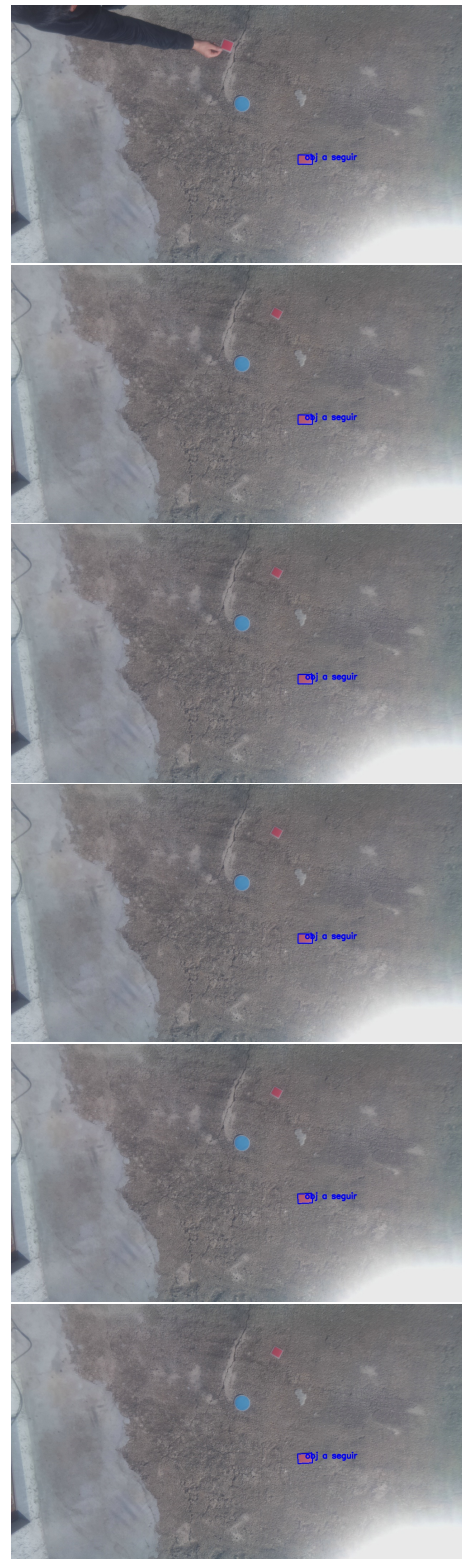
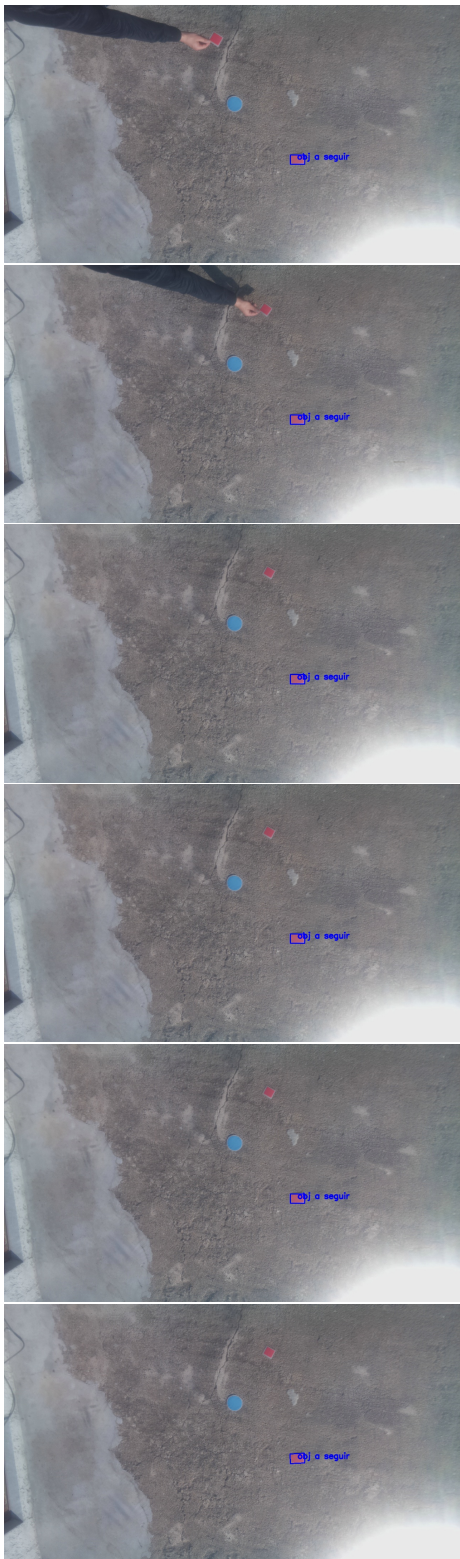
	Camera Module v1	Camera Module v2
Net price	\$25	\$25
Size	Around 25 × 24 × 9 mm	
Weight	3g	3g
Still resolution	5 Megapixels	8 Megapixels
Video modes	1080p30, 720p60 and 640 × 480p60/90	1080p30, 720p60 and 640 × 480p60/90
Linux integration	V4L2 driver available	V4L2 driver available
C programming API	OpenMAX IL and others available	OpenMAX IL and others available
Sensor	OmniVision OV5647	Sony IMX219
Sensor resolution	2592 × 1944 pixels	3280 × 2464 pixels
Sensor image area	3.76 × 2.74 mm	3.68 × 2.76 mm (4.6 mm diagonal)
Pixel size	1.4 µm × 1.4 µm	1.12 µm × 1.12 µm
Optical size	1/4"	1/4"

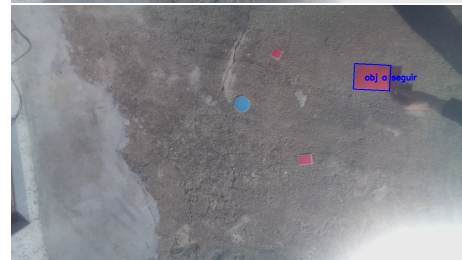
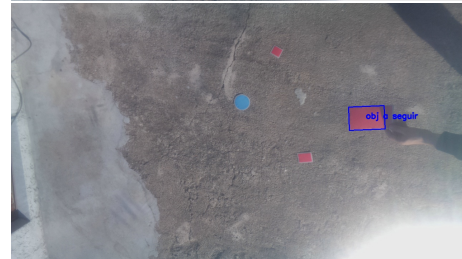
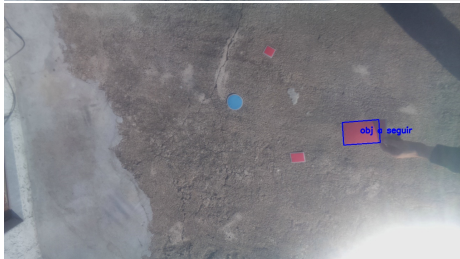
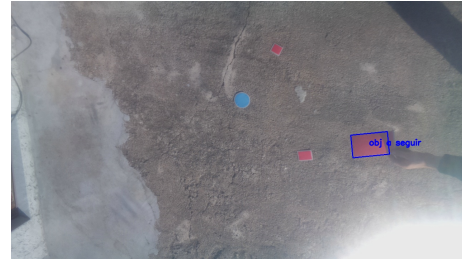
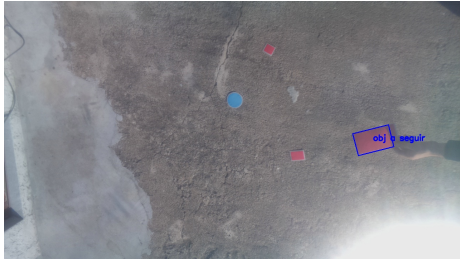
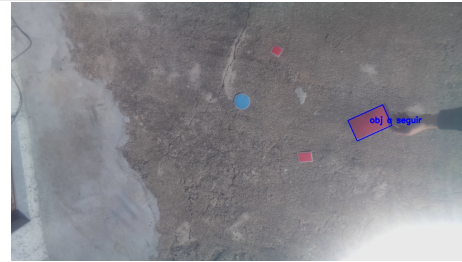
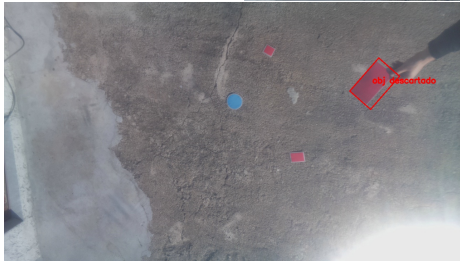
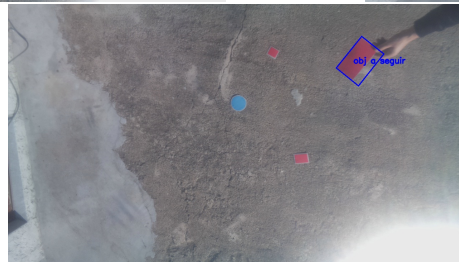
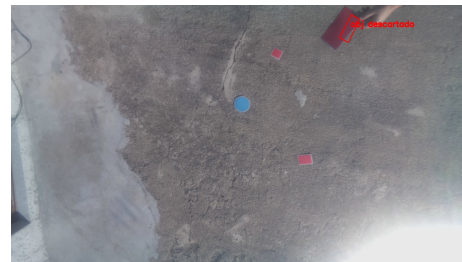
Capítulo 7

Anexo B

7.1 Escolha dos objetos por maior área

Localização do objeto	Localização anterior	Distância de Movimento	Distância de <i>Tracking</i>
(819, 431)	(819, 431)	192,5668715	279,0072662
(819, 431)	(819, 431)	192,5668715	279,0072662
(820, 431)	(819, 431)	193,49677	280,3545822
(820, 430)	(820, 431)	193,1320792	279,8261871
(819, 431)	(820, 430)	192,5668715	279,0072662
(820, 431)	(819, 431)	193,49677	280,3545822
(819, 431)	(820, 431)	192,5668715	279,0072662
(820, 431)	(819, 431)	193,49677	280,3545822
(820, 430)	(820, 431)	193,1320792	279,8261871
(819, 431)	(820, 430)	192,5668715	279,0072662
(965, 165)	(819, 431)	379,0118732	549,1446465
(1000, 335)	(965, 165)	360,8670115	522,8548272
(1031, 389)	(1000, 335)	392,0739726	568,0701274
(998, 402)	(1031, 389)	360,4552677	522,2582577
(995, 362)	(998, 402)	355,0056338	514,3623645
(989, 322)	(995, 362)	351,0626725	508,6494667
(991, 250)	(989, 322)	367,8328425	532,9475157
(986, 209)	(991, 250)	377,5142381	546,9747454
(819, 432)	(986, 209)	192,9378138	279,5447189
(821, 431)	(819, 432)	194,4273643	281,7029064
(820, 431)	(821, 431)	193,49677	280,3545822
(820, 431)	(820, 431)	193,49677	280,3545822
(820, 431)	(820, 431)	193,49677	280,3545822
(820, 431)	(820, 431)	193,49677	280,3545822
(819, 432)	(820, 431)	192,9378138	279,5447189
(820, 431)	(819, 432)	193,49677	280,3545822
(820, 431)	(820, 431)	193,49677	280,3545822
(820, 431)	(820, 431)	193,49677	280,3545822
(819, 432)	(820, 431)	192,9378138	279,5447189
(820, 432)	(819, 432)	193,8659331	280,8894571
(818, 432)	(820, 432)	192,0104164	278,2010266



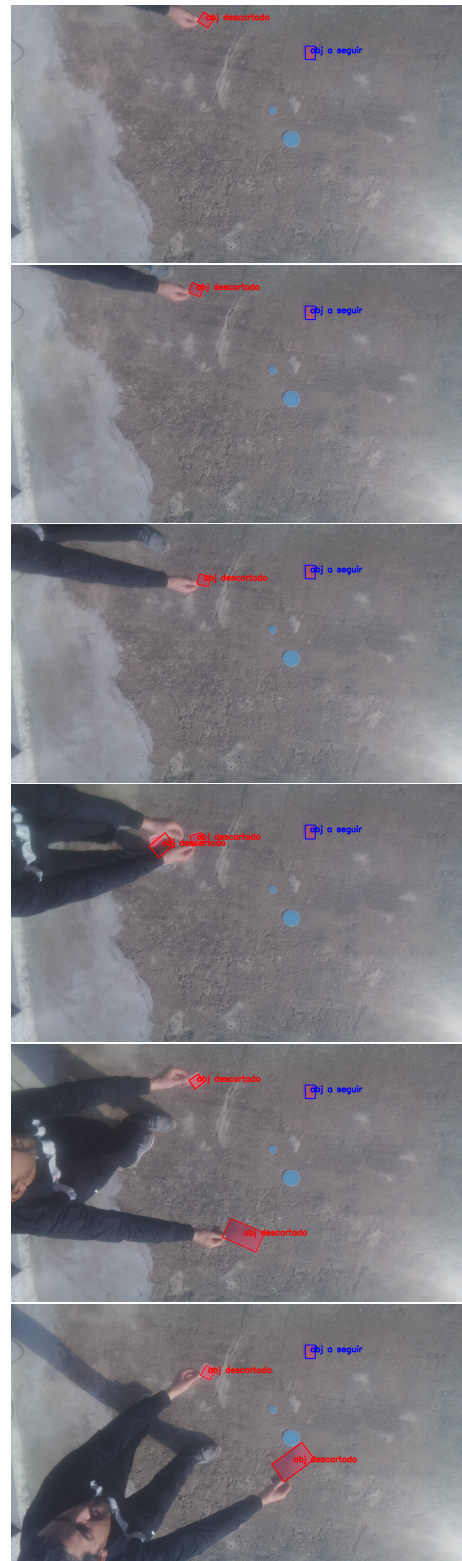
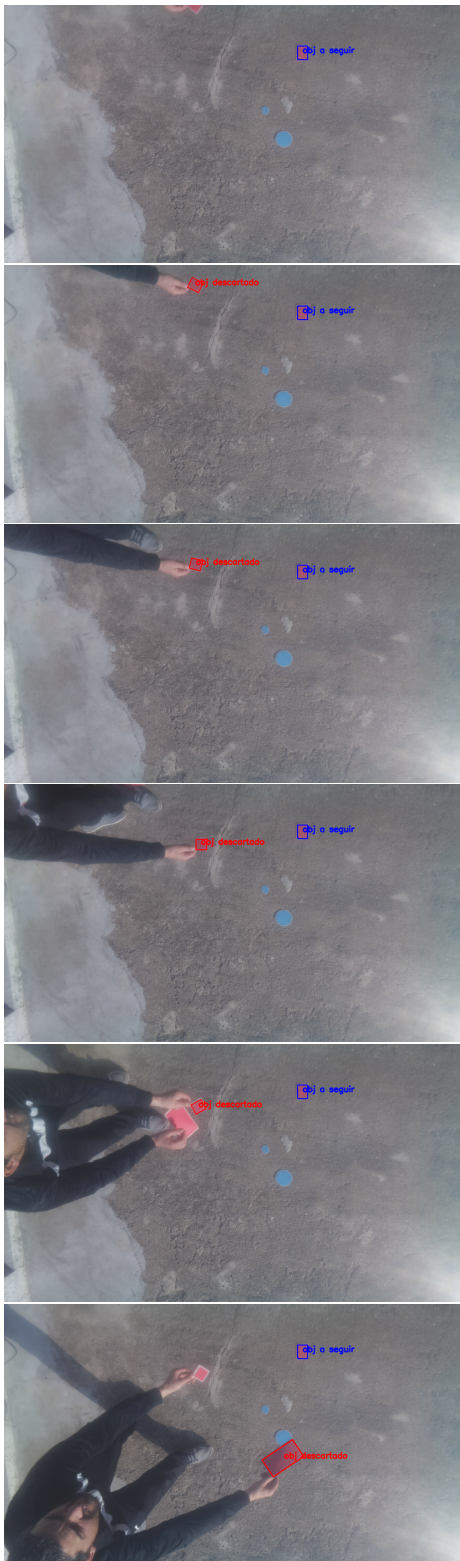


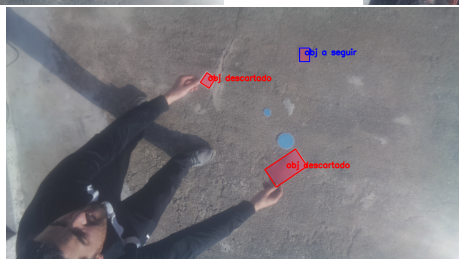
Capítulo 8

Anexo C

8.1 Escolha dos objetos pelo utilizador

Localização do objeto	Localização anterior	Distância de Movimento	Distância de <i>Tracking</i>
(834, 133)	(834, 133)	298,6050904	580,2718843
(543, 41)	(834, 133)	333,421655	580,2718843
(534, 55)	(543, 41)	322,8947197	580,2718843
(516, 68)	(534, 55)	317,2380809	580,2718843
(536, 112)	(516, 68)	268,923781	580,2718843
(834, 133)	(536, 112)	298,6050904	580,2718843
(834, 133)	(834, 133)	298,6050904	580,2718843
(833, 133)	(834, 133)	297,9563726	579,0112471
(834, 133)	(833, 133)	298,6050904	580,2718843
(518, 104)	(834, 133)	283,5842027	580,2718843
(834, 133)	(518, 104)	298,6050904	580,2718843
(834, 133)	(834, 133)	298,6050904	580,2718843
(834, 133)	(834, 133)	298,6050904	580,2718843
(834, 133)	(834, 133)	298,6050904	580,2718843
(834, 133)	(834, 133)	298,6050904	580,2718843
(834, 133)	(834, 133)	298,6050904	580,2718843
(834, 133)	(834, 133)	298,6050904	580,2718843
(834, 133)	(834, 133)	298,6050904	580,2718843
(833, 132)	(834, 133)	298,7189314	580,4931088
(834, 133)	(833, 132)	298,6050904	580,2718843
(836, 142)	(834, 133)	293,155249	569,6813423
(735, 229)	(836, 142)	161,8208886	314,4625292
(496, 547)	(735, 229)	236,019067	314,4625292
(757, 252)	(496, 547)	159,2262541	309,4204402
(756, 261)	(757, 252)	152,502459	296,3542555
(753, 262)	(756, 261)	149,5760676	290,6674715
(743, 263)	(753, 262)	141,4849815	274,9442641
(746, 267)	(743, 263)	141,0141837	274,0293743
(749, 260)	(746, 267)	147,9222769	287,4537009
(768, 235)	(749, 260)	178,9105922	347,6725273





Bibliografia

- [1] Riham Altawy and Amr M Youssef. Security, privacy, and safety aspects of civilian drones: A survey. *ACM Transactions on Cyber-Physical Systems*, 1(2):7, 2017.
- [2] Ludovic Apvrille, Tullio Tanzi, and Jean-Luc Dugelay. Autonomous drones for assisting rescue services within the context of natural disasters. In *General Assembly and Scientific Symposium (URSI GASS), 2014 XXXIth URSI*, pages 1–4. IEEE, 2014.
- [3] Dario Floreano and Robert J Wood. Science, technology and the future of small autonomous drones. *Nature*, 521(7553):460–466, 2015.
- [4] David Forsyth and Jean Ponce. *Computer vision: a modern approach*. Upper Saddle River, NJ; London: Prentice Hall, 2011.
- [5] G. Guglieri G. Ristorto, F. Mazzetto and F. Quagliotti. Monitoring performances and cost estimation of multirotor unmanned aerial systems in precision farming*. In *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*.
- [6] Sai Ram Ganti and Yoohwan Kim. Implementation of detection and tracking mechanism for small uas. In *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*.
- [7] Seokwoo Jung Hanseob Lee and David Hyunchul Shim. Vision-based uav landing on the moving vehicle. In *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*.
- [8] Antonio JR Neves, Manuel Camarneiro, and Lucas Cozinheiro. Vehicle detection on low altitude images based on edge density. *SIGNAL 2017 Editors*, page 40, 2017.
- [9] António JR Neves, Armando J Pinho, Daniel A Martins, and Bernardo Cunha. An efficient omnidirectional vision system for soccer robots: From calibration to object detection. *Mechatronics*, 21(2):399–410, 2011.

- [10] Antonio JR Neves, Alina Trifan, Bernardo Cunha, and José Luís Azevedo. Real-time color coded object detection using a modular computer vision library. *Advances in Computer Science: an International Journal*, 5(1):110–123, 2016.
- [11] raspberrypi.org. Camera module, 2015.
- [12] Paul J Springer. *Military robots and drones: a reference handbook*. ABC-CLIO, 2013.
- [13] Vineetha Shrinivasan Victor Wolfe, William Frobe and Tsung-Yen Hsieh. Detecting and locating cell phone signals from avalanche victims using unmanned aerial vehicles. In *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*.
- [14] Antidio Viguria, Ivan Maza, and Anibal Ollero. Distributed service-based cooperation in aerial/ground robot teams applied to fire detection and extinguishing missions. *Advanced Robotics*, 24(1-2):1–23, 2010.
- [15] Hanxuan Yang, Ling Shao, Feng Zheng, Liang Wang, and Zhan Song. Recent advances and trends in visual tracking: A review. *Neurocomputing*, 74(18):3823–3831, 2011.
- [16] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *Acm computing surveys (CSUR)*, 38(4):13, 2006.
- [17] Chunhua Zhang and John M Kovacs. The application of small unmanned aerial systems for precision agriculture: a review. *Precision agriculture*, 13(6):693–712, 2012.